CRISP-DM Artefacts
NextMeal Recommendation System

Business Understanding

Objective

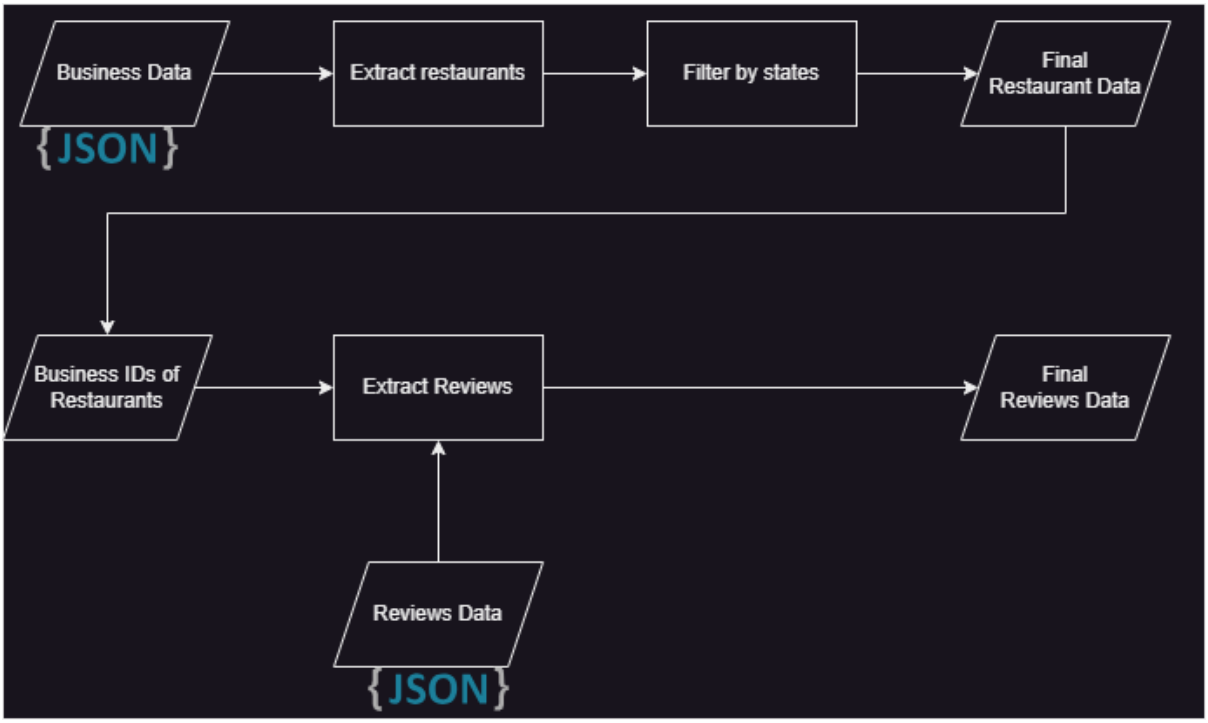
The project focuses on analyzing restaurants and reviews data to gain insights and make predictions or recommendations.

Business Goal

Understanding customer preferences and improving recommendations in the restaurant domain to increase customer retention and reservations.

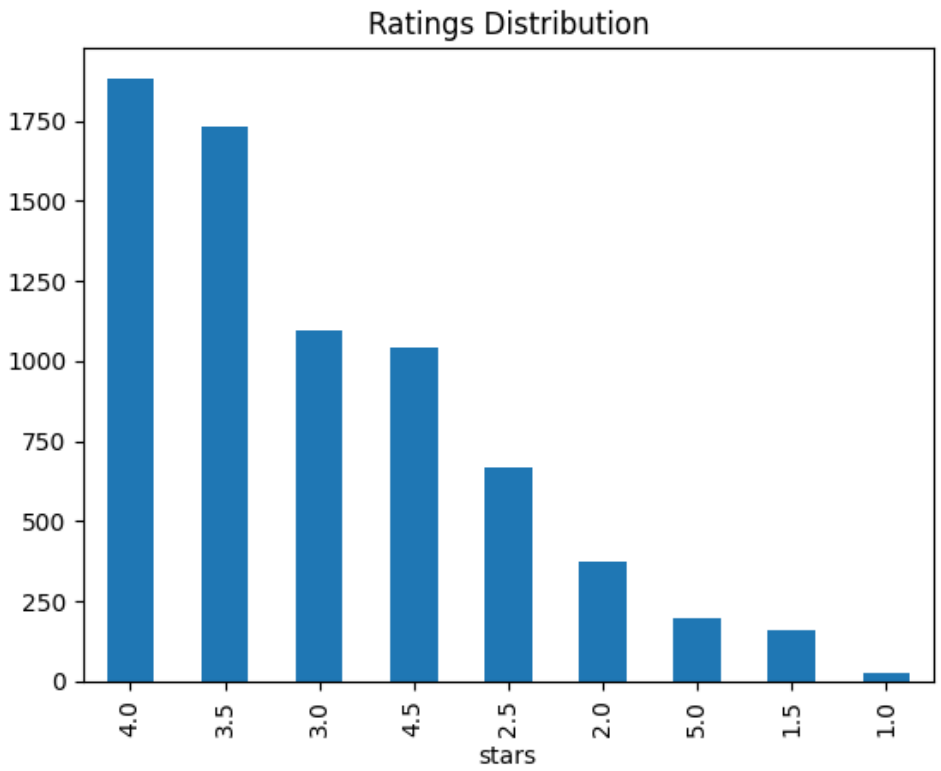
Data Understanding

We have used the Yelp dataset, specifically the businesses and reviews data. Since our use case only involves restaurants, we performed some preprocessing to extract relevant data. We filtered the JSON file for restaurants as well as for selected states ('CA', 'AZ', and 'NJ'). Then we used the same set of business IDs for filtering reviews corresponding to the restaurants to form our final dataset.

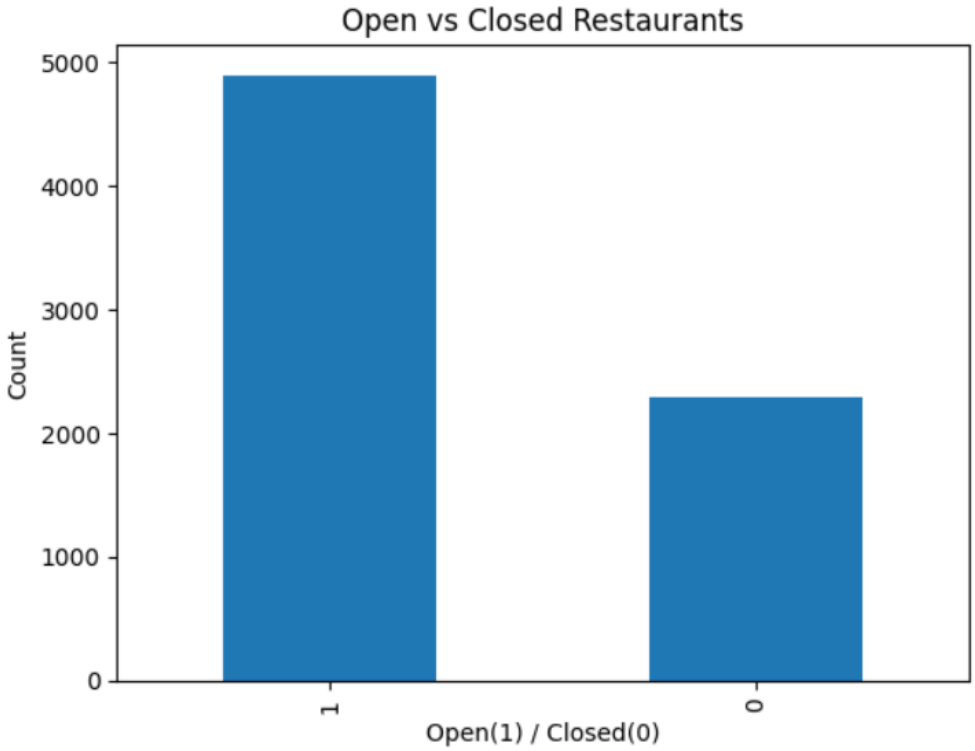


Preprocessing pipeline

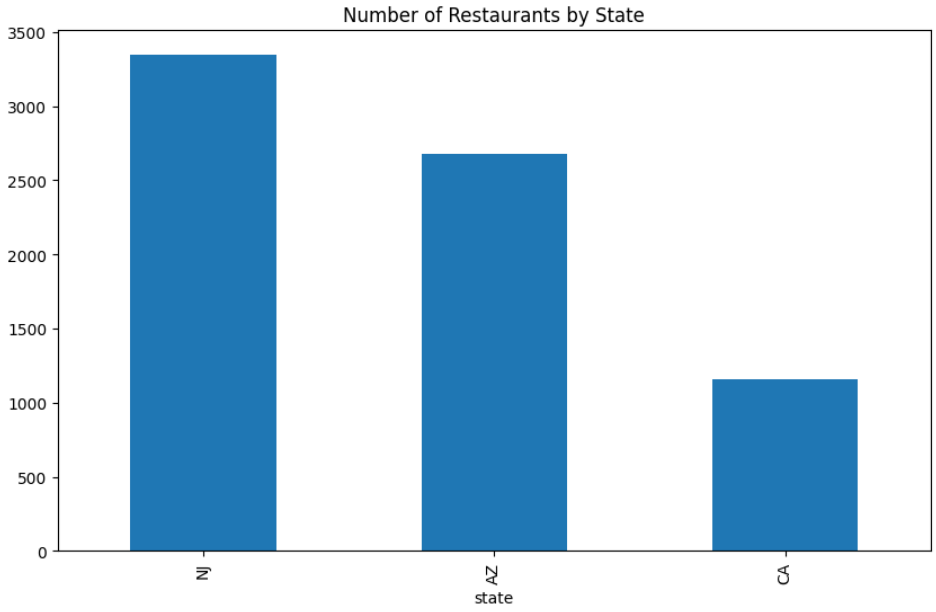
With our dataset ready, we performed some initial exploratory data analysis by creating visualizations.



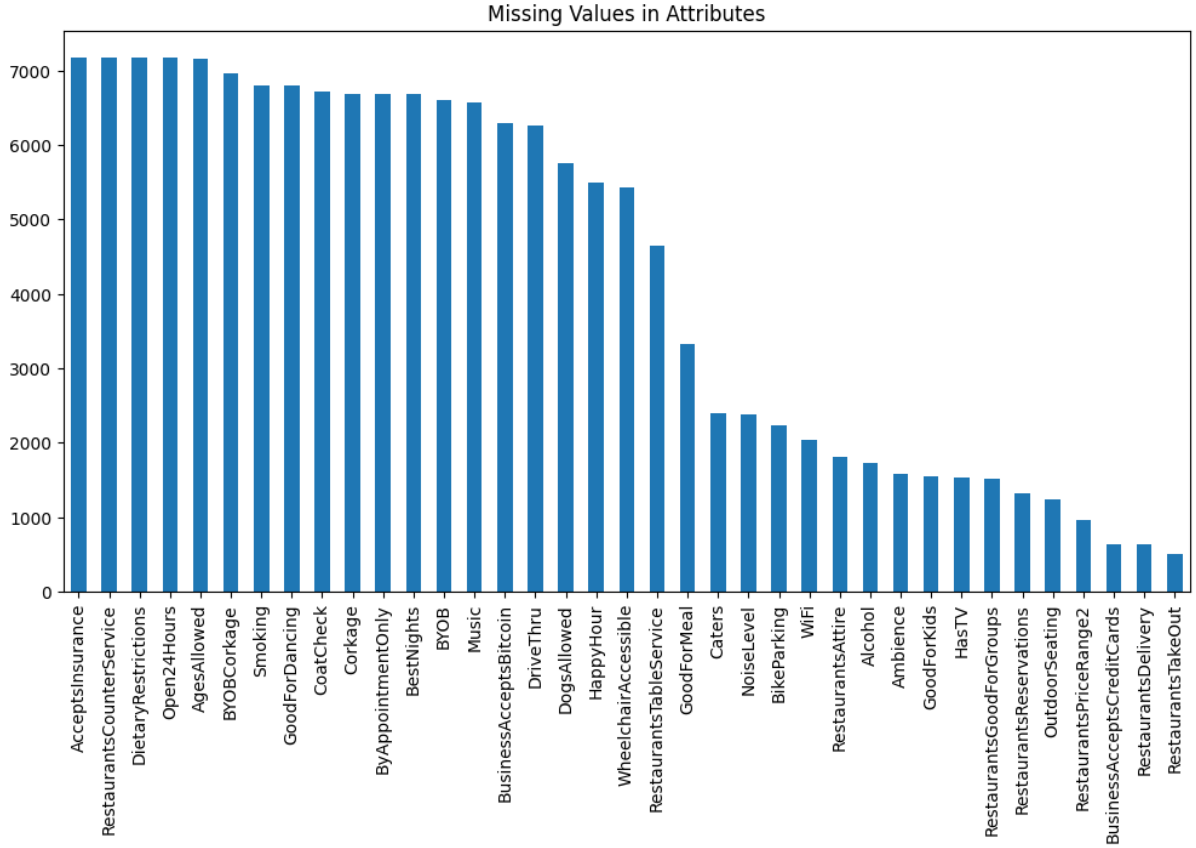
Histogram of ratings in the restaurant dataset



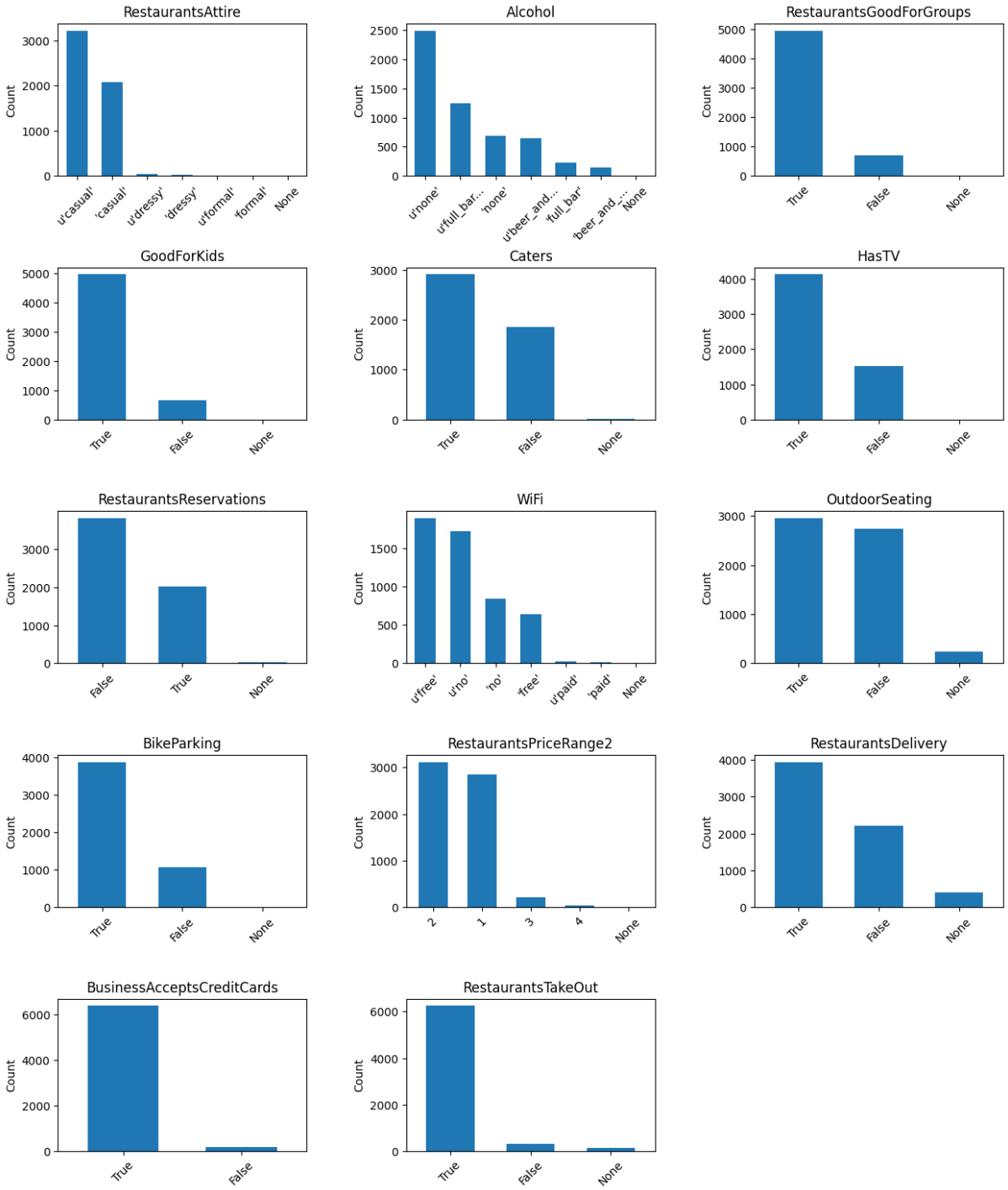
Histogram of Open/Closed restaurants



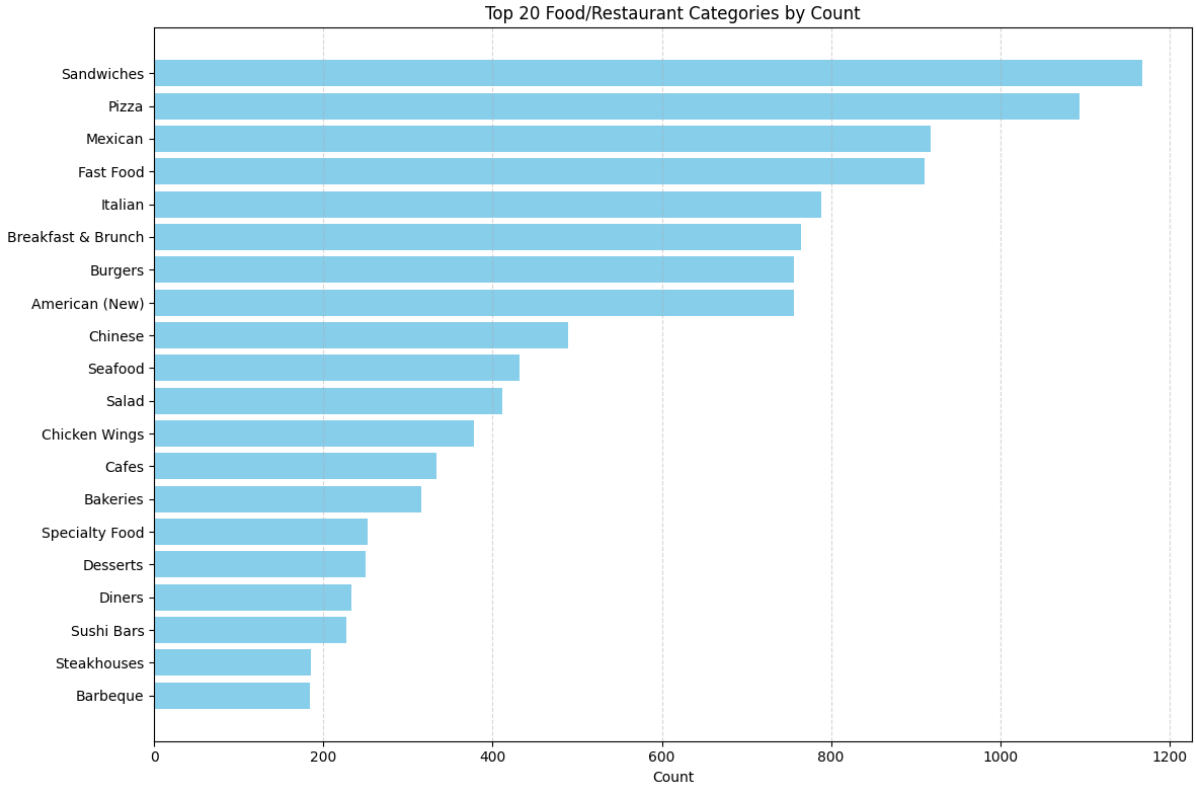
Restaurant by State



Missing value in Attributes column



Attributes histograms



Top 20 Categories

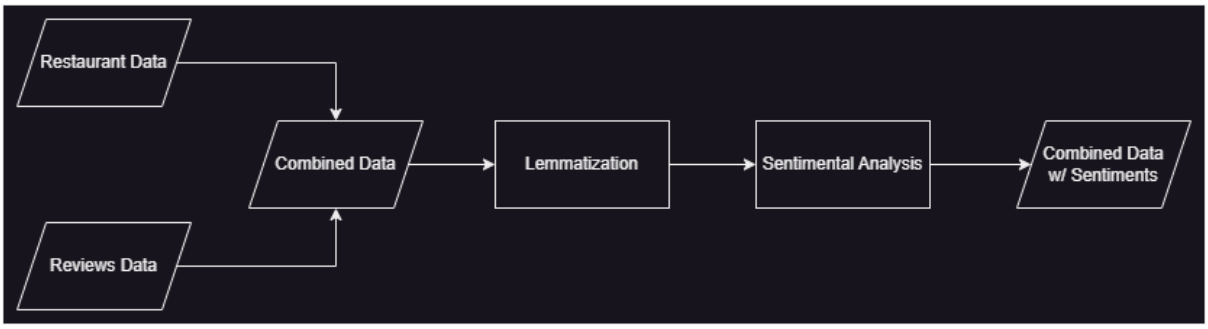
Having explored the restaurants data, we have observed a few features which were complicated. Attributes feature is stored as key-value pairs inside a JSON. For analysis, we need to flatten the data. Same thing applies to Categories column, which contains comma-separated values in text format. Both of these are highlighted as the main areas of focus required in the Data Preparation phase.

Data Preparation

For the Attributes feature, we flattened the internal JSON data into a binary feature vector, similar to One-Hot Encoding. As these attributes weren't consistently available in all features, we selected the top 15 most frequently appearing ones to avoid creating a sparse matrix.

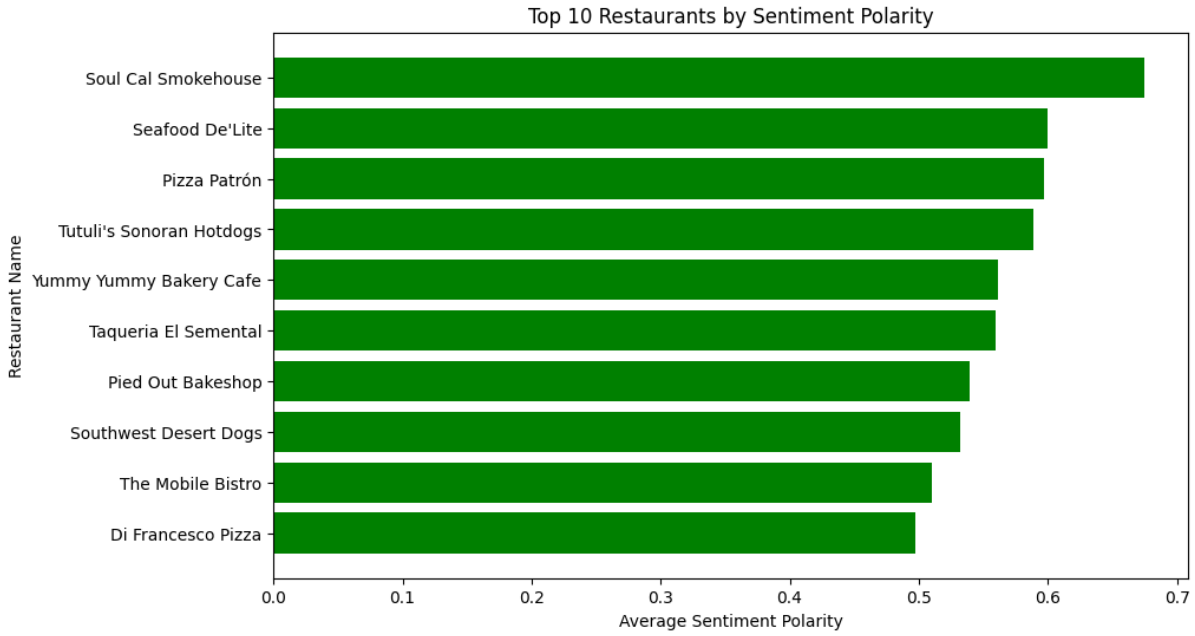
For the Columns feature, we flattened the text data into a binary feature vector as well, which came out to about 700 fields. As these were way too many and the result would be sparse again, we performed feature selection by choosing the top 20 most frequently appearing categories.

With this data made consistent, we joined it with Reviews data for sentimental analysis. After joining Restaurants and Reviews into a combined dataframe, we performed Lemmatization and Sentimental Scoring using the nltk and Text Blob libraries. Each review went through this pipeline shown below, generating a sentimental polarity score based on the refined text data.

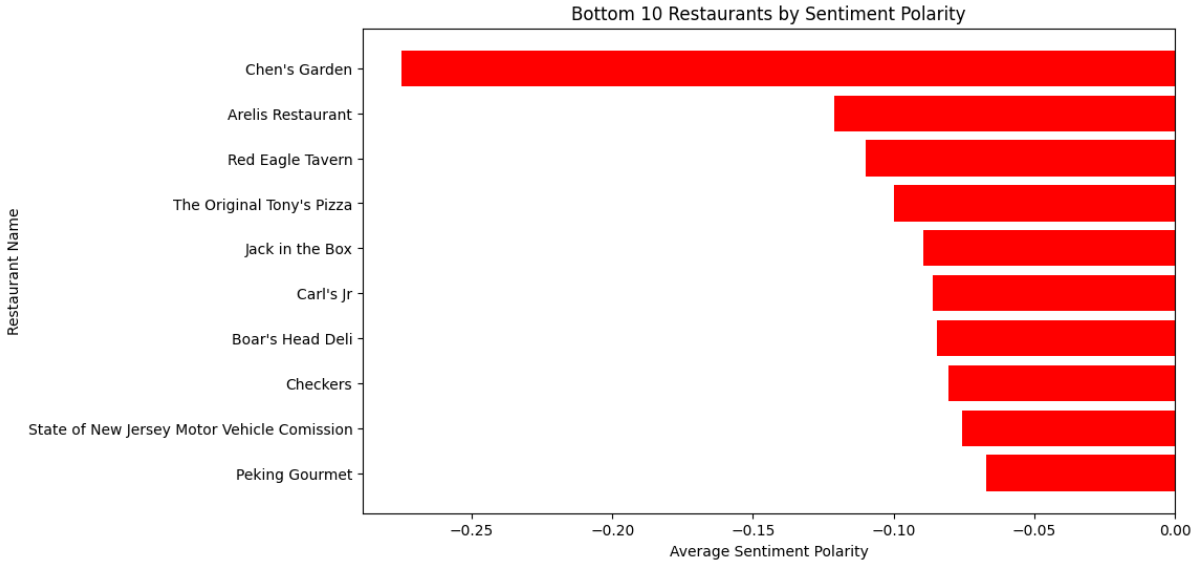


Sentimental Analysis pipeline

Using these scores, we did some extra analysis by plotting the best and worst 10 restaurants.



Top 10 Restaurants by Sentiment Polarity



Bottom 10 Restaurants by Sentiment Polarity

Modeling

For our recommendation system, we went with three techniques/model:

1. Collaborative Filtering
2. Content-based Filtering
3. Hybrid model

For the first, we created an interaction matrix between the users and restaurants and using Singular Value Decomposition (SVD), we predicted ratings for all combinations. This allowed us to generate the Top N recommendations for the users.

For the second, we selected relevant features (from attributes and categories) to make recommendations by finding the most similar restaurants. We used the Cosine similarity function for this process to rank the similar restaurants for the user.

Finally, for the hybrid model, we calculated a weighted average of predictions from both models to capture all features and ratings appropriately.

Evaluation

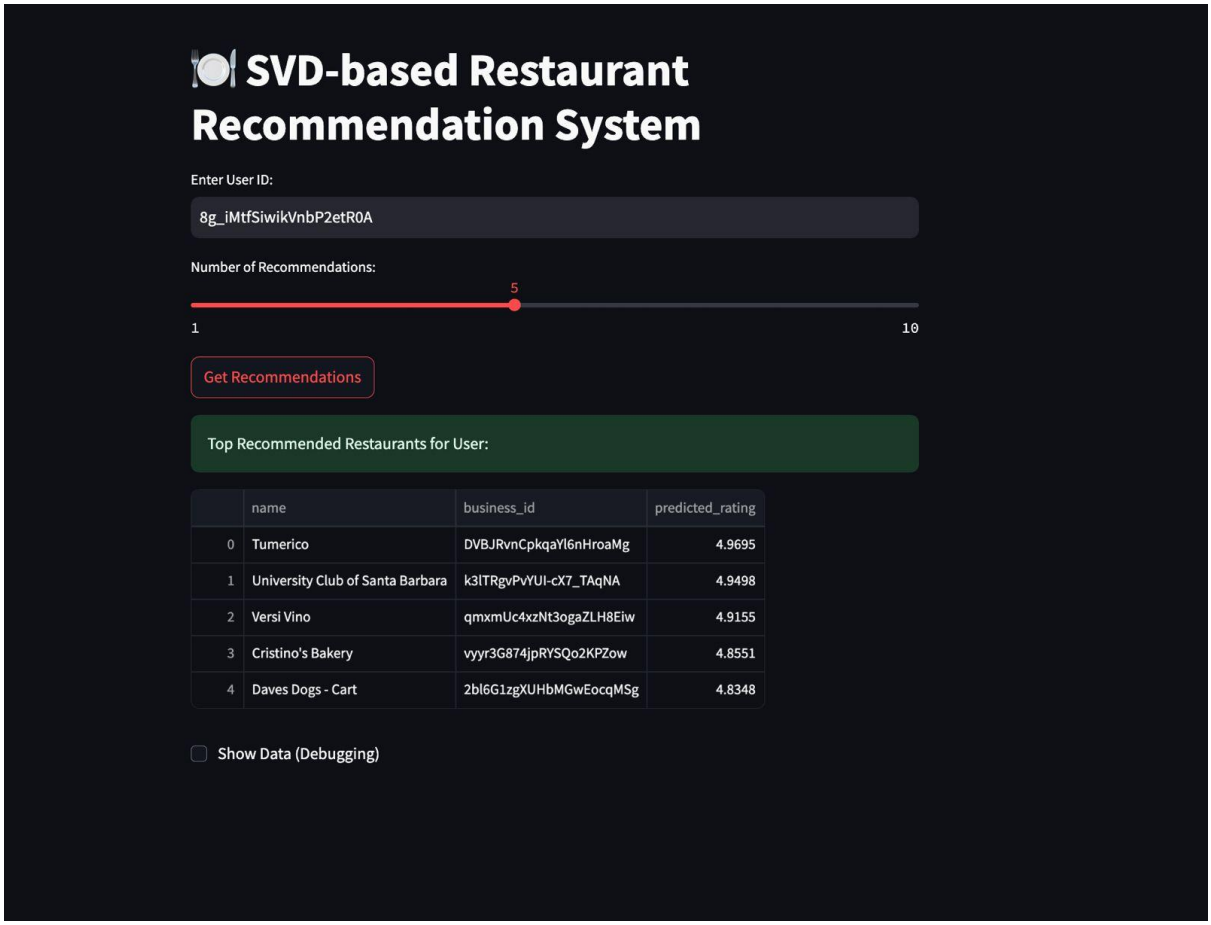
We used the Root Mean Square Error (RMSE) for evaluating the three models. It was observed that Collaborative Filtering performed the best within our dataset based on selected metrics, whereas the Content-based and Hybrid models were slightly worse, largely due to a sparse matrix creation from the selected features skewing the performance.

Model	**RMSE**
Collaborative Filtering	1.29
Content-based Filtering	1.37
Hybrid Model	1.38

Although the hybrid model is considered the best option over a larger volume and period of time, with the current data, we chose the Collaborative Filtering model for now due to marginally better metrics.

Deployment

After saving the model we chose as a PKL file, we integrated it with a Streamlit application to create a full-stack experience where given a user id and required number of recommendations, the model will generate the result and display their expected ratings for the given user.


🍽 SVD-based Restaurant Recommendation System
Enter User ID:
8g_iMtfSiwikVnbP2etR0A
Number of Recommendations:
5
1
10
Get Recommendations
Top Recommended Restaurants for User:
name
business_id
predicted_rating
0 Tumerico DVBJRvnCpkqaYl6nHroaMg 4.9695
1 University Club of Santa Barbara k3lTRgvPvYUI-cX7_TAqNA 4.9498
2 Versi Vino qmxmUc4xzNt3ogaZLH8Eiw 4.9155
3 Cristino's Bakery vyyr3G874jpRYSQo2KPZow 4.8551
4 Daves Dogs - Cart 2bl6G1zgXUHbMGwEocqMSg 4.8348
Show Data (Debugging)